# OBJECTIVE:

* Understanding the operating modes of a timer:
* Understanding how to use a timer for creating delays and generating pulses.

# REFERENCES:

* Lab manual Chapter 4, 5
* Atmel-2505-Setup-and-Use-of-AVR-Timers\_ApplicationNote\_AVR130.pdf

# EXPERIMENT 1:

1. Write a delay subroutine of 1 ms using Timer 0. Use this subroutine to generate a 500 Hz pulse on pin PA0.

| LDI R16,(1<<0) ;R16=0x20  SBI DDRA,0 ;PB5 as an output  LDI R17,0  OUT PORTA,R17  MAIN:  CALL DELAY\_HALF\_PERIOD ; Call the delay subroutine to generate half the period  SBI PORTA, 0 ; Set PA0 high  CALL DELAY\_HALF\_PERIOD ; Call the delay subroutine to generate half the period  CBI PORTA, 0 ; Set PA0 low  RJMP MAIN ; Repeat the process    DELAY\_HALF\_PERIOD:  LDI R20, 0x1F ; Load R20 with a suitable value for the half period delay  CALL DELAY ; Call the delay subroutine  RET  DELAY:  LDI R20,0x7D  OUT TCNT0,R20 ;load timer0  LDI R20,0x0  OUT TCCR0A,R20  LDI R20,0x03  OUT TCCR0B,R20 ;Normal mode, PRE64. clk  AGAIN:  SBIS TIFR0,TOV0 ;if TOV0 is set skip next  RJMP AGAIN  LDI R20,0x0  OUT TCCR0B,R20 ;stop Timer0  LDI R20,(1<<TOV0) ;R20 = 0x01  OUT TIFR0,R20 ;clear TOV0 flag  RET  RJMP MAIN |
| --- |

1. Simulate and modify the program to achieve accurate pulse generation.
2. Connect pin PA0 to an oscilloscope to verify.

# EXPERIMENT 2:

1. Write a program to generate a 64 us square wave using Timer 0 in Normal mode. Use pin OC0 as the output.

| ; Replace with your application code  LDI R16,(1<<3) ;R16=0x08  SBI DDRB,3 ;PB3 as an output  LDI R17,0  OUT PORTB,R17  MAIN:  CALL DELAY  EOR R17,R16 ;toggle D5 of R17  OUT PORTB,R17 ;toggle PB5  RJMP MAIN  DELAY:  LDI R20,0xF8  OUT TCNT0,R20 ;load timer0 delay = (256 + 1 - x) \*64 \* 0.125us = 64us  LDI R20,0x0  OUT TCCR0A,R20  LDI R20,0x03  OUT TCCR0B,R20 ;Normal mode, PRE64. clk  AGAIN:  SBIS TIFR0,TOV0 ;if TOV0 is set skip next  RJMP AGAIN  LDI R20,0x0  OUT TCCR0B,R20 ;stop Timer0  LDI R20,(1<<TOV0) ;R20 = 0x01  OUT TIFR0,R20 ;clear TOV0 flag  RET  RJMP MAIN |
| --- |

1. Write a program to generate a 64 us square wave using Timer 1 in CTC mode. Use pin OC0 as the output.

| Replace with your application code  LDI R16,HIGH(RAMEND)  OUT SPH,R16  LDI R16,LOW(RAMEND)  OUT SPL,R16    SBI DDRB,3 ;PB5 as an output  BEGIN:SBI PORTB,3 ;PB5 = 1  RCALL DELAY\_64us  CBI PORTB,3 ;PB5 = 0  RCALL DELAY\_64us  RJMP BEGIN    DELAY\_64us:  LDI R20,0x00  STS TCNT1H,R20 ;TEMP = 0  STS TCNT1L,R20 ;TCNT1L = 0, TCNT1H = TEMP    LDI R20,0x00  STS OCR1AH,R20 ;TEMP = 0x00 delay = (x+1) \* 0.125us = 64us => x = 7  LDI R20,0x07  STS OCR1AL,R20 ;OCR1AL = 0x07, OCR1AH = TEMP    LDI R20,0x00  STS TCCR1A,R20 ;WGM11:10=00  LDI R20,0x0B  STS TCCR1B,R20 ;WGM13:12=01,CS=CLK rescaler64,ctc  AGAIN:  SBIS TIFR1,OCF1A ;if OCF1A is set skip next instruction  RJMP AGAIN  LDI R19,0  STS TCCR1B,R19 ;stop timer  STS TCCR1A,R19 ;  LDI R20,1<<OCF1A  OUT TIFR1,R20 ;clear OCF1A flag  RET |
| --- |

1. Connect pin OC0 to an oscilloscope and observe.

# EXPERIMENT 3:

1. Given the program for generating two PWM pulses on OC0A and OC0B, connect pins OC0A and OC0B to two oscilloscope channels, measure and record the waveform, and explain the obtained waveform.

| .org 00  call initTimer0  start:  rjmp start  initTimer0:  // Set OC0A (PB3) and OC0B (PB4) pins as outputs  ldi r16, (1 << PB3) | (1 << PB4);  out DDRB,r16  ldi r16, (1 << COM0B1)|(1 << COM0A1) | (1 << WGM00)|(1 << WGM01)  out TCCR0A,r16 // setup TCCR0A  ldi r16, (1 << CS01)  out TCCR0B,r16 // setup TCCR0B  ldi r16, 100  out OCR0A,r16 //OCRA = 100  ldi r16, 75  out OCR0B,r16 //OCRB = 75  ret |
| --- |

# EXPERIMENT 4:

1. Modify the program for different combinations of TCCR0A and TCCR0B registers as described in the table:

|  | TCCR0A | | | | | | | | TCCR0B | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | COM0A1 | COM0A0 | COM0B1 | COM0B0 |  |  | WGM01 | WGM00 | FOC0A | FOC0B |  |  | WGM02 | CS02 | CS01 | CS00 |
| 2 | 1 | 0 | 1 | 0 |  |  | 1 | 1 |  |  |  |  | 0 | 0 | 1 | 0 |
| 3 | 1 | 0 | 1 | 0 |  |  | 1 | 1 |  |  |  |  | 1 | 0 | 1 | 0 |
| 4 | 1 | 0 | 1 | 0 |  |  | 0 | 1 |  |  |  |  | 0 | 0 | 1 | 0 |

1. Connect pins OC0A and OC0B to two oscilloscope channels, measure and record the waveforms, and explain the obtained results.

# EXPERIMENT 5:

1. Write a program to generate a 1 kHz square wave with a duty cycle of 25% on pin OC0B.
2. Connect to an oscilloscope and measure the output waveform.
3. Connect OC0B to the R channel of an RGB LED. Write a program to increase the duty cycle on OC0B from 0% to 100% and then decrease it back to 0% over 10 ms, with a 1% increment.

# EXPERIMENT 1:

1. Answer the following questions:
   1. What is the maximum delay achievable using Timer 0 with an 8 MHz frequency? Explain the calculation.

𝑇𝐷𝐿 = 𝑛 ∗ 𝑁 ∗ 𝑇𝑜𝑠𝑐

với n max khi cài đặt n=0 -> 𝑛𝑚𝑎𝑥=256 (8bit)

N là hệ số chia -> 𝑁𝑚𝑎𝑥 = 1024

f=8MHz, T=1/f -> 𝑇𝑜𝑠𝑐 = 0,125 ∗ 10−6 (𝑠)

TDLmax = 𝑛𝑚𝑎𝑥 ∗ 𝑁𝑚𝑎𝑥 ∗ 𝑇𝑜𝑠𝑐 = 0,032768 (𝑠)

* 1. What is the maximum delay achievable using Timer 1 with an 8 MHz frequency? Explain the calculation.
  2. Explain how to calculate the prescaler values and the values loaded into Timer0 registers for this experiment.
  3. Source code for the program with comments.

| ........................................................................................................................ |
| --- |

# EXPERIMENT 2:

1. Answer the following questions:
   1. In Normal mode, when is the TOVx bit set to 1?
   2. In CTC mode, when is the OCFx bit set to 1?
   3. Provide the register configurations for Timer 0 for both cases.
   4. Source code for the programs in both cases.

| ........................................................................................................................ |
| --- |

# EXPERIMENT 3:

1. Answer the following questions:
   1. Describe the waveform on the oscilloscope (capture and insert it).
   2. Explain the reasons for the observed waveform (frequency, duty cycle, phase).

# EXPERIMENT 4:

1. Answer the following questions:
   1. Identify the working modes of Timer 0 corresponding to the values in the table.
2. Capture images of the waveforms corresponding to the different working modes and explain the results.

# EXPERIMENT 5:

1. Answer the following questions:
   1. In which mode is Timer 0 operating?
   2. What values are loaded into Timer 0 registers, and why?
2. Present the source code with comments.

| ........................................................................................................................ |
| --- |